

Professional MCDU box meets Open serious MOD

Combining HW MCDU unit to FlyByWire MCDU-server

Hisao Munakata (magu775<at>gmail.com)

Hobbyist (Not representing any organization)

2022-6-9

Disclaimer

I will introduce various SW/HW components, however

- I am not representing any party who develops these components.
- Things introduced here is just a quick hack, and cannot be the reference.
- Implementation highly depends on my environment, may not portable.
- Of course, no guarantee, no support commitment, play with your own risk
- Component (product) name introduced here are registered trade mark of each vendors

Microsoft Flight Simulator 2020

Microsoft Flight Simulator 2020 (a.k.a. MSFS or MSFS2020)

Product brief summary

- Runs on Windows PC and Xbox
- Highly utilize resources on cloud
- Revived by Asobo Studio
- Includes various planes includes flight systems
- Can use 3rd party HW (joy-stick, rudder,...)
- Also, accept 3rd party plane model (as MOD)



Now MSFS becomes entire flight simulator execution platform

3rd party gaming device (common products)

Standard Windows game device

- USB HID class device
- Windows OS support natively
- Test program integrated to Control Panel
- No extra driver needed
- MSFS detects and works
- Mostly right out of the box experience



A320 PRO-MX MCDU

Flightdeck Solutions E-Series A320 PRO-MX MCDU

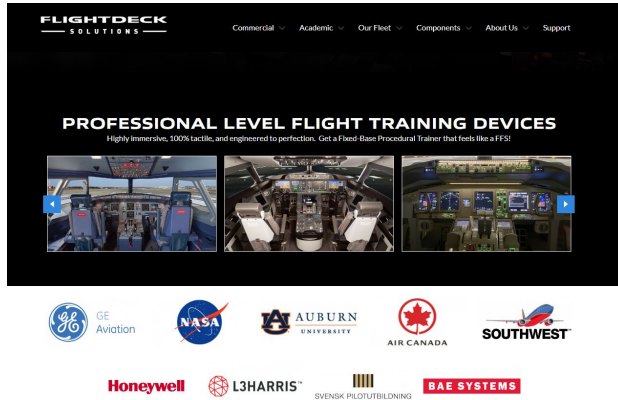


Multi-Function Control Display Unit

- Emulate Airbus MCDU HW
- Identical size, material, color
- HDMI interface (video input)
- Ethernet port (event input/output)
- Speak standard TCP/IP protocol
- Does not contain Flight Management and Guidance SW

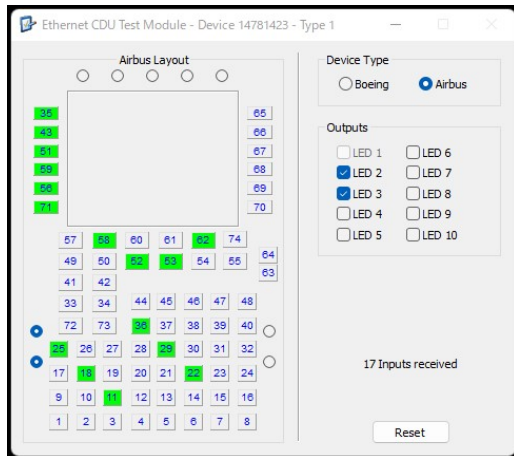
<https://flightdecksolutions.com/components/p/fds-a320-pro-mx-mcd-u-e>

Flight training equipment provider for professional use



<https://flightdecksolutions.com/>

Bundled HW test tool (= interface IT module)



Bundled HW check tool

- HW validation tool
- Developed by **TEKWork Limited**
- **Generate TCP/IP packet**
- Key input event
 - MCDU to PC (tool)
- Indicator on/off control
 - PC (tool) to MCDU
- **MSFS support is not ready**

Capture TCP/IP traffic (using Wireshark)

(ip.host == 192.168.11.12) && tcp && !(tcp.analysis.keep_alive)						
No.	Time	Source	Destination	Protocol	Length	Info
20	2.188188	192.168.11.12	192.168.11.37	TCP	67	10346 → 61510 [PSH, ACK] Seq=1 Ack=1 Win=1500 Len=13
21	2.228294	192.168.11.37	192.168.11.12	TCP	54	61510 → 10346 [ACK] Seq=1 Ack=14 Win=65231 Len=0
27	2.417388	192.168.11.12	192.168.11.37	TCP	68	10346 → 61510 [PSH, ACK] Seq=14 Ack=1 Win=1500 Len=14
28	2.463092	192.168.11.37	192.168.11.12	TCP	54	61510 → 10346 [ACK] Seq=1 Ack=28 Win=65217 Len=0
39	3.092811	192.168.11.12	192.168.11.37	TCP	67	10346 → 61510 [PSH, ACK] Seq=28 Ack=1 Win=1500 Len=13
40	3.133247	192.168.11.37	192.168.11.12	TCP	54	61510 → 10346 [ACK] Seq=1 Ack=41 Win=65204 Len=0
41	3.269194	192.168.11.12	192.168.11.37	TCP	68	10346 → 61510 [PSH, ACK] Seq=41 Ack=1 Win=1500 Len=14
42	3.319752	192.168.11.37	192.168.11.12	TCP	54	61510 → 10346 [ACK] Seq=1 Ack=55 Win=65190 Len=0
68	3.931465	192.168.11.12	192.168.11.37	TCP	67	10346 → 61510 [PSH, ACK] Seq=55 Ack=1 Win=1500 Len=13
69	3.977029	192.168.11.37	192.168.11.12	TCP	54	61510 → 10346 [ACK] Seq=1 Ack=68 Win=65177 Len=0
72	4.108003	192.168.11.12	192.168.11.37	TCP	68	10346 → 61510 [PSH, ACK] Seq=68 Ack=1 Win=1500 Len=14
73	4.149377	192.168.11.37	192.168.11.12	TCP	54	61510 → 10346 [ACK] Seq=1 Ack=82 Win=65163 Len=0
> Frame 27: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface \Device\NPF_{45FBDC29-4493-46F5-8485-9827478B502D}, id 0 > Ethernet II, Src: Microchi_e1:8b:ef (68:27:19:e1:8b:ef), Dst: WistronI_d6:11:7a (98:ee:cb:d6:11:7a) > Internet Protocol Version 4, Src: 192.168.11.12, Dst: 192.168.11.37 > Transmission Control Protocol, Src Port: 10346, Dst Port: 61510, Seq: 14, Ack: 1, Len: 14 > Data (14 bytes)						
0000	98 ee cb d6 11 7a 68 27	19 e1 8b ef 08 00 45 00zh'E.			
0010	00 36 06 f4 00 00 64 06	b8 4c c0 a8 0b 0c c0 a8	-6....d- .L.....			
0020	0b 25 28 6a f0 46 05 9e	ca ce 6a 26 0a dc 50 18	-%(j·F· ..j&...P·			
0030	05 dc 1b ae 00 00 42 31	3d 53 57 3a 34 34 3a 4fB1 =SW:44:0			
0040	46 46 0d 0a		FF..			

FryByWire Simulations A32NX project

Open Source Project who develops MOD for MSFS

The screenshot shows the FlyByWire website. The top navigation bar includes links for NOTAMS, Projects, Documentation, Map, and Community. The main content area is divided into two sections. The left section, titled 'Community Insights', features a blue header and a paragraph describing the community. Below this, four statistics are displayed: 251 Live Flights, 3656 Commits, 186 Contributors, and 1M+ Downloads. A Discord section follows, with a blue header and a paragraph about the server. A 'Join the Community' button is located at the bottom of this section. The right section features a map of Europe with various cities marked by blue pins. The bottom of the website has a dark blue footer with logos for Flightsim.to, FSNews, YOUR CONTROLS, and Salty Simulations, along with a small green button with a white arrow pointing up.

flybywire

NOTAMS Projects Documentation Map Community

Community Insights

Discover the extensive community behind every FlyByWire Simulations aircraft - a vibrant and active online group that prioritises collaborative work and openness.

251	3656	186	1M+
Live Flights	Commits	Contributors	Downloads

Discord

Our Discord server is where we plan the entirety of our projects and provide most of our support. Join us to chat with other members of the community, get started with contributing, or ask us a question!

[Join the Community →](#)

Flightsim.to FSNews YOUR CONTROLS Salty Simulations

<https://flybywiresim.com/>

Fully comply with Open Source development model

flybywiresim / a32nx Public

Product Team Enterprise Explore Marketplace Pricing

Search Sign in Sign up

Sponsor Notifications Fork 848 Star 4.3k

Code Issues 312 Pull requests 32 Discussions Actions Projects 3 Wiki Security Insights

master 25 branches 85 tags Go to file Code About

3 authors fix show yellow gs reference line in correct conditions (#7274) ✓ dets68 15 hours ago 3,656 commits

.ace	feat(efb): flyPadOS 3 (#6528)	10 days ago
.cargo	build: cargo workspace at the project root (#3346)	16 months ago
.github	fix: show yellow gs reference line in correct conditions (#7274)	15 hours ago
.vscode	feat(efb): flyPadOS 3 (#6528)	10 days ago
docs	feat(efb): flyPadOS 3 (#6528)	10 days ago
flybywire-sircraft-a320-neo	refactor(mcdul): improved departure/arrival pages (#7259)	yesterday
jest	feat: failures (#5359)	10 months ago
mfs-avionics-mirror	fix(mcdul): f-pin page annotations (#7057)	2 months ago
scripts	build: use synchronous copyFile call in build script (#7243)	11 days ago
src	fix: show yellow gs reference line in correct conditions (#7274)	15 hours ago
typings	feat(efb): flyPadOS 3 (#6528)	10 days ago
.clang-format	feat: reworked Fly-By-Wire, Autopilot, Autothrust, Engine Model, PFD ...	14 months ago
.editorconfig	feat: reworked Fly-By-Wire, Autopilot, Autothrust, Engine Model, PFD ...	14 months ago
.eslintignore	fix(build): add missing eslint ignores (#7030)	2 months ago
.gitignore	infotool: add support to RCEC simulator framework (RCECF3)	3 months ago

The A32NX Project is a community driven open source project to create a free Airbus A320neo in Microsoft Flight Simulator that is as close to reality as possible.

flybywiresim.com

react javascript rust typescript reactjs matlab a320 mfs2020

Readme
GPL-3.0 license
Code of conduct
4.3k stars
162 watching
848 forks

Releases 63

v0.8.1 (Latest) 24 days ago

+ 62 releases

<https://github.com/flybywiresim/a32nx>

Rolling release mode (Stable / Development / Experimental)

READY TO FLY?

Download

We have included many options to download our addons, you can use our custom and simple installer to always keep your products up to date, or you can download using standalone installations.

- Integrates seamlessly into Microsoft Flight Simulator - no external programs required.
- Safe, trustworthy, and constantly updated to assure nothing is broken.
- One click install, neatly organized into one compact folder.

Installer

Our easy-to-use installer is the easiest way to get started with our addons. Simply launch and install any addon you want, with only two clicks.

[Download](#)

Direct Download

If you prefer a direct download, the following links are available.

Stable Release	Download
Development Build	Download
Experimental Build	Download

<https://flybywiresim.com/a32nx/>

A32NX MOD aims to reproduce precise reality

FlyByWire A32NX Overview

This section of the FlyByWire Documentation is dedicated to the A32NX add-on itself. It covers the software and more technical aspects of the FlyByWire add-on.

The A32NX Project is a community-driven open source project to create a free Airbus A320neo in Microsoft Flight Simulator that is as close to reality as possible. It started out as an enhancement project to the default A320neo and is now proceeding as an independent add-on project aiming to bring the FlyByWire A32NX up to payware-level systems depth and functionality, all for free.

Aircraft Configuration(s)

The following aircraft configurations are currently simulated:

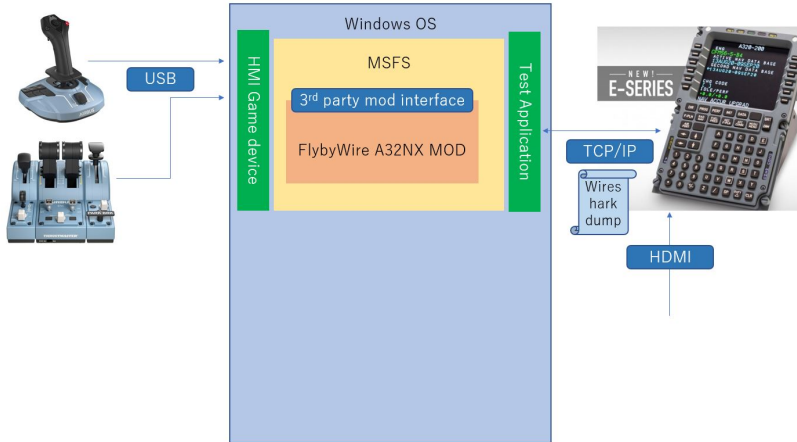
Simulated Hardware	
Model	A320-251N
Engine	CFM LEAP 1A-26
APU	APS3200
FMS	Honeywell Release H3
FWC Std.	H2F9C
TAWS	Honeywell EGPWS
ACAS	Honeywell TPA-100B
ATC	Honeywell TRA-100B
MMR	Honeywell 1MMR
WXR	Honeywell RDR-4000

<https://docs.flybywiresim.com/fbw-a32nx/>

A32NX reality includes

- Look & Feel (color, sound,...)
- Physical motion model
- **Flight management system**
 - FMS algorithm
 - **MCDU operations**
 - SimBrief flight planner
- integrate real world
 - Weather
 - Routing (via Navigraph)
- Follows every MSFS updates

System Architecture (original)

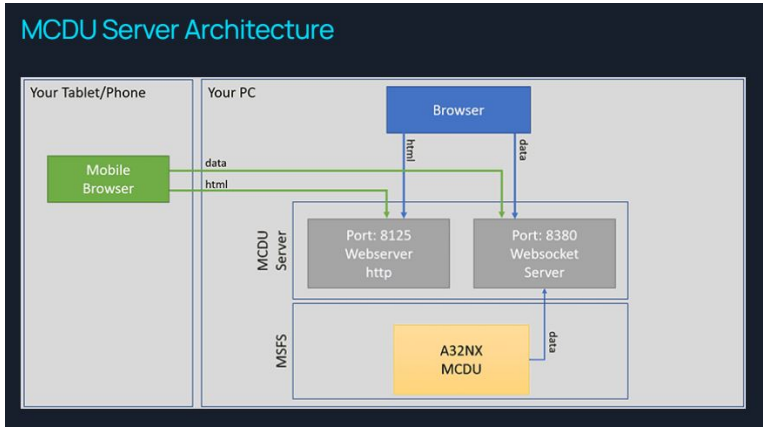


Newly introduced **MCDU web interface**

MCDU runs as Web application

- Can run MCDU on separate tablet, smartphone,..
- Can use real printer
- Runs **MCDU-server** to interface to MSFS/A32NX MOD
- Use **WebSocket** protocol
- `https://docs.flybywiresim.com/fbw-a32nx/feature-guides/web-mcdu/`

MCDU server architecture (Web socket interface)

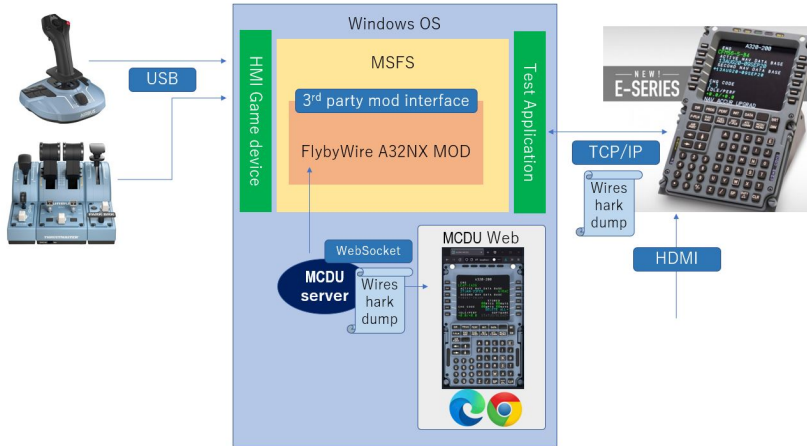


<https://docs.flybywiresim.com/fbw-a32nx/feature-guides/web-mcdu/#starting-the-mcdu-server>

Capture the web socket data (using Wireshark)

tcpstream eq 0						
No.	Time	Source	Destination	Protocol	Length	Info
74	1.798965	192.168.11.41	192.168.11.37	TCP	78	[TCP Dup ACK 73#1] 54239 → 8380 [ACK] Seq=14 Ack=1675 Win=4043 Len=0 TSval=175558256 TSecr=42423677 SLE=227 SRE=1675
90	1.989937	192.168.11.41	192.168.11.37	TCP	79	54239 → 8380 [PSH, ACK] Seq=14 Ack=1675 Win=4096 Len=13 TSval=175558456 TSecr=42423677
91	1.990191	192.168.11.37	192.168.11.41	TCP	75	8380 → 54239 [PSH, ACK] Seq=1675 Ack=27 Win=8191 Len=9 TSval=42423909 TSecr=175558456
92	1.994849	192.168.11.41	192.168.11.37	TCP	66	54239 → 8380 [ACK] Seq=27 Ack=1684 Win=4095 Len=0 TSval=175558460 TSecr=42423909
129	2.266320	192.168.11.37	192.168.11.41	TCP	1733	8380 → 54239 [PSH, ACK] Seq=1684 Ack=27 Win=8191 Len=1667 TSval=42424185 TSecr=175558460
131	2.297945	192.168.11.37	192.168.11.41	TCP	1514	[TCP Retransmission] 8380 → 54239 [PSH, ACK] Seq=1903 Ack=27 Win=8191 Len=1448 TSval=42424217 TSecr=175558460
132	2.301967	192.168.11.41	192.168.11.37	TCP	66	54239 → 8380 [ACK] Seq=27 Ack=3351 Win=4043 Len=0 TSval=175558768 TSecr=42424185
133	2.301967	192.168.11.41	192.168.11.37	TCP	78	[TCP Dup ACK 132#1] 54239 → 8380 [ACK] Seq=27 Ack=3351 Win=4043 Len=0 TSval=175558768 TSecr=42424217 SLE=1903 SRE=3351
162	2.637012	192.168.11.41	192.168.11.37	TCP	79	54239 → 8380 [PSH, ACK] Seq=27 Ack=3351 Win=4096 Len=13 TSval=175559097 TSecr=42424217
163	2.637294	192.168.11.37	192.168.11.41	TCP	75	8380 → 54239 [PSH, ACK] Seq=3351 Ack=40 Win=8191 Len=9 TSval=42424556 TSecr=175559097
164	2.638894	192.168.11.41	192.168.11.37	TCP	66	54239 → 8380 [ACK] Seq=40 Ack=3360 Win=4095 Len=0 TSval=175559105 TSecr=42424556
165	2.889284	192.168.11.37	192.168.11.41	TCP	1735	8380 → 54239 [PSH, ACK] Seq=3360 Ack=40 Win=8191 Len=1669 TSval=42424808 TSecr=175559105
166	2.917471	192.168.11.41	192.168.11.37	TCP	66	54239 → 8380 [ACK] Seq=40 Ack=5029 Win=4043 Len=0 TSval=175559383 TSecr=42424808
177	3.154683	192.168.11.41	192.168.11.37	TCP	60	[TCP Keep-Alive] 54239 → 8380 [ACK] Seq=39 Ack=5029 Win=4096 Len=0
178	3.154716	192.168.11.37	192.168.11.41	TCP	66	[TCP Keep-Alive ACK] 8380 → 54239 [ACK] Seq=5029 Ack=40 Win=8191 Len=0 TSval=42425074 TSecr=175559383
195	3.571729	192.168.11.41	192.168.11.37	TCP	79	54239 → 8380 [PSH, ACK] Seq=40 Ack=5029 Win=4096 Len=13 TSval=175560032 TSecr=42425074
196	3.572220	192.168.11.37	192.168.11.41	TCP	75	8380 → 54239 [PSH, ACK] Seq=5029 Ack=53 Win=8191 Len=9 TSval=42425491 TSecr=175560032
197	3.573972	192.168.11.41	192.168.11.37	TCP	66	54239 → 8380 [ACK] Seq=53 Ack=5038 Win=4095 Len=0 TSval=175560040 TSecr=42425491
> Frame 163: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface \Device\NPF_{45F8DC29-4A93-46F5-8485-98274788502D}, id 0 > Ethernet II, Src: WistronI_d6:11:7a (98:ee:c8:d6:11:7a), Dst: 1e:18:c4:b7:28:8e (1e:18:c4:b7:28:8e) > Internet Protocol Version 4, Src: 192.168.11.37, Dst: 192.168.11.41 > Transmission Control Protocol, Src Port: 8380, Dst Port: 54239, Seq: 3351, Ack: 40, Len: 9 > Data (9 bytes) Data: 81076576656e743a33 [Length: 9]						
0000	1e 18 c4 b7 28 8e 98 ee	cb d6 11 7a 08 00 45 00(---z--E-			
0010	00 3d e5 ee 40 00 80 06	00 00 c0 a8 0b 25 c0 a8	..-@---%..			
0020	0b 29 20 bc d3 df a7 e8	62 8c ab 48 61 00 80 18	.)b..Ha...			
0030	1f ff 97 ce 00 00 01 01	08 0a 02 87 58 ec 0a 76X...v			
0040	d1 b9 81 07 65 76 65 6e	74 3a 33even t:3			

System Architecture (w/MCDU web interface)



System Architecture (integrated with translator)

